

原创经典，句句珠玑

Broadview
www.broadview.com.cn

资深动画师乔珂力作，ActionScript 3.0经典图书横空出世
蓝色理想、闪吧、DDC传媒三大网站大力推荐

ActionScript 3.0 权威指南

ActionScript 3.0: The Definitive Guide

乔珂 等编著



- ◎ 彻底理清ActionScript 3.0的概念、语法和新特性，一一瓦解学习的难点
- ◎ 遵循ActionScript 3.0自身逻辑，深入剖析动画设计的各种算法
- ◎ 深入ActionScript 3.0的深层机制，建立独立、完整的开发思维
- ◎ 结合实际开发，提供大量典型示例和3个综合案例，具有超强的实用性
- ◎ 适用于Flash、Flex等多种开发环境，具有较强的通用性



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

ActionScript 3.0 权威指南

ActionScript 3.0: The Definitive Guide

乔珂 等编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书将 ActionScript 3.0 作为一种全新的、独立的、具有自己特色的语言，系统全面地介绍了此语言的结构和概念。

全书以培养实际开发能力为目的，将语言的基本语法、核心概念及新增功能统一起来。根据知识点在语言逻辑体系和实际应用中的权重，对语法和概念重新进行分级和组织。结合 Flash 网络应用和脚本开发的实际，围绕面向对象的开发思想，深入阐述了 ActionScript 3.0 的自身逻辑。不仅解决了应该怎么做的问题，还回答了为何如此的疑问。引领读者体会语法表象之内的深层美感，培养 ActionScript 3.0 开发思维，使读者能流畅地编写代码。

本书内容由浅入深平缓过渡，适用于各种程度的读者。即使读者没有相关知识背景，也能轻松阅读本书，并最终掌握 ActionScript 3.0。对于有开发经验的读者，阅读本书有助于脱离惯性思维，重新认识这种语言。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

ActionScript 3.0 权威指南 / 乔珂等编著. —北京：电子工业出版社，2008.7
ISBN 978-7-121-06852-2

I. A… II. 乔… III. 动画—设计—图形软件, Flash ActionScript 3.0—指南 IV. TP391.41-62

中国版本图书馆 CIP 数据核字（2008）第 081200 号

责任编辑：高洪霞

印 刷：北京东光印刷厂

装 订：三河市皇庄路通装订厂

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：787×1092 1/16 印张：52.5 字数：1294 千字

印 次：2008 年 7 月第 1 次印刷

印 数：5000 册 定价：89.00 元（含光盘 1 张）

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

序：美妙的语言

PREFACE

欢迎阅读《ActionScript 3.0 权威指南》一书，这将为您提供一个了解 ActionScript 3.0 的绝佳机会。本书将带您进入一个充满创意的世界，一个让您能够充分发挥想象力的世界。

ActionScript 最初设计用于 Flash 交互控制动画，但它的发展远远超出了这个范围。如今，它已成为一个强大的平台，用于创建各种类型的交互式内容。从简单的动画到复杂的交互式应用程序，ActionScript 3.0 都能胜任。本书将带您了解 ActionScript 3.0 的各个方面，包括其语法、数据类型、控制结构、类、事件、动画、网络、安全、性能优化等。

一本令人震撼之作，凝结作者多年心血！

一本平易近人之作，每位 ActionScript 爱好者都可从中受益！

一本值得反复玩味，仔细揣摩之作，带你领悟 ActionScript 的至高境界！

在本书中，您将了解到 ActionScript 3.0 的许多新特性，包括：新的数据类型、新的控制结构、新的类系统、新的动画系统、新的网络系统、新的安全系统、新的性能优化技术等。这些新特性将使您能够创建出更加强大、更加复杂的交互式内容。本书将为您提供详细的代码示例，帮助您理解这些新特性的用法。此外，本书还将为您提供一些实用的技巧和最佳实践，帮助您提高开发效率。本书适合所有对 ActionScript 3.0 感兴趣的读者阅读，无论是初学者还是经验丰富的开发人员。希望您能从本书中获得宝贵的知识和经验，为您的 ActionScript 3.0 开发之旅开启新的篇章。

现在，让我们一起开始 ActionScript 的神奇之旅吧！

在本书中，您将了解到 ActionScript 3.0 的许多新特性，包括：新的数据类型、新的控制结构、新的类系统、新的动画系统、新的网络系统、新的安全系统、新的性能优化技术等。这些新特性将使您能够创建出更加强大、更加复杂的交互式内容。本书将为您提供详细的代码示例，帮助您理解这些新特性的用法。此外，本书还将为您提供一些实用的技巧和最佳实践，帮助您提高开发效率。本书适合所有对 ActionScript 3.0 感兴趣的读者阅读，无论是初学者还是经验丰富的开发人员。希望您能从本书中获得宝贵的知识和经验，为您的 ActionScript 3.0 开发之旅开启新的篇章。

感谢您对本书的支持，希望您能从本书中获得宝贵的知识和经验，为您的 ActionScript 3.0 开发之旅开启新的篇章。

序：美妙的语言

PREFACE

借阅读《ActionScript 3.0 权威指南》书稿的机会，我不禁重新审视 ActionScript——这门独具魅力的语言，一个广阔思想的展台。

ActionScript 最初设计用于 Flash 交互与控制动画。当开发者可以快速地建立动画，配合音效，并根据浏览者的意图来产生互动时，是多么令人振奋！越来越多的人开始使用 Flash 创建属于自己的个人网站，在互联网上展示自己。网络上随处可见五彩缤纷的 Flash 作品、游戏，使我看到了许多让人赞叹的精彩创意。

自 2.0 以后 ActionScript 类库的更新，标志着 ActionScript 已经成为一门面向对象语言，随着越来越多的 Flash 作为前端 RIA (Rich Internet application) 应用的出现，宣告了 ActionScript 进入新的时代。直到 Micromedia 被 Adobe 公司收购以后，ActionScript 遇到了一个不大不小的变革。说大，是因为语言结构发生了很大的变化；说小，是因为虽然类的组织进行了更加完善的架构，而根本的思想还是从之前的体系演化而来的。当然，在这个过程中，ActionScript 的能力得到了进一步的强化，也吸引了很多从事 Java 或 C++ 的程序员进来，于是，Flash 被更加明确地分成了两类：界面动画图形设计与交互程序。不可否认，ActionScript 3 核心的类库做出了大幅度的重构，并且在语法上也和 ActionScript 2 存在着较大的分歧。然而，要注意的是，虽然这个改动导致了很多从事 ActionScript 2 开发者难以适应，但是，新的架构体系与其说是改变了很多东西，还不如说是提炼了很多东西，它以更规范、更高效的形态来逐步改变开发者的思想。令人欣慰的是，本书作者乔珂也清楚看到了 ActionScript 的变化脉络，在这部 800 多页的专著中，他用易于理解的思路和语言详细阐述了 ActionScript 作为一门日趋丰满的面向对象语言的独特魅力，并结合自身开发经验为读者提供了丰富的案例来借鉴学习。

我一直认为，无论是 Flash IDE 也好，Flex Builder 也好，或者什么都好，作为开发环境而言，并没有优劣之分。直到现在，还有很多朋友在提出类似到底哪一个更好的问题。其实，无论是 RIA 何种应用，游戏，网站，动画展示等，无非都是以“swf”为后缀的文件，用 Flash Player 解析呈现而已，这是 Flash 在终端所展示的形态。现在，你既可以用 Flash 轻松加入喜爱的图片、音乐，并有序地组织好这些资源，也可以使用 Flex 快速创建应用程序，通过调用别的后台语言，如 PHP、JSP、ColdFusion 等获取数据，再使用合理的逻辑把数据和界面组织起来。如果你熟悉了 ActionScript，在这个平台里，你可以尽情地按照你的想象来发挥，我相信现在的 ActionScript 3 有着非常优秀的架构体系和广泛的应用面，也相信会有更多的朋友加入到这个充满活力的阵营中来。

感谢 Adobe 开发组的所有成员，奉献和创造了如此美妙的语言。

希望看到本书的读者，不要错过领悟美妙的 ActionScript 的机会。

“蓝色理想”网站站长 曾沐阳

www.blueidea.com

前言

PREFACE

一本概念阐述与实例讲解俱佳的好书，值得你反复阅读！

学 ActionScript 的难点是什么

ActionScript 是一门开发语言，但是使用它的并不是经过训练的程序员，大部分都是从设计方面转过来的，因此使用者对 ActionScript 的很多概念都很陌生，在懵懵懂懂的状态下就开始编程了。这种情况下，很容易遇到瓶颈：

- 代码冗长，不易阅读，后期维护非常困难。
- Bug 很多，经常出现一些莫名其妙的错误。
- 不能使用面向对象的思路设计程序，编写了大量的重复代码。

.....

之所以出现这些问题，就是因为没有深刻理解 AS 中的各种概念。我从 Flash 5.0 就开始编写 AS 代码，其中走了很多弯路，同时也积累下了大量的笔记和各种注意事项。后来总结的一条就是：**如果理解了语言的核心概念和内部机制，不仅能开发出健壮的 Flash 应用程序，而且能发现 ActionScript 3.0 自身的独特美感。**

市场上虽然关于 ActionScript 的书很多，但是都是流于介绍语法，而没有从本质、从概念上对其进行阐述，因此我才萌生了写书的想法，希望能对读者有所帮助。

本书的内容安排

本书在安排内容时，将 ActionScript 3.0 作为全新、独立、特色的语言看待。

- 本书将 ActionScript 3.0 作为全新的语言，而不是早期版本的强化升级。以培养实际开发能力为目的，将语言的基本语法、核心概念及新增功能统一起来。根据知识点在语言逻辑体系和实际应用中的权重，对语法和概念重新进行分级和组织。真正做到了由浅入深的平缓过渡。即使读者没有相关知识背景，也能毫无障碍地阅读本书，并最终掌握 ActionScript 3.0。
- 本书将 ActionScript 3.0 作为独立的语言，而不是类似语言的简化变种。根据 Flash 网络应用和脚本开发的实际，深入阐述了语言的自身逻辑。不仅解决了应该如何的问题，还回答了为何如此的疑问。帮助有开发经验的读者脱离惯性思维，重新认识语言，培养 ActionScript 3.0 开发思维。
- 本书将 ActionScript 3.0 作为特色的语言，而不是语句概念的机械组合。围绕面向对象的开发思想，阐述了 OOP 在语言中的具体实现。尊重 ActionScript 自身逻辑，详细讨论了编译执行、事件交互、显示列表、脚本动画等特色概念。引领读者体会 ActionScript 3.0 语法表象之内的深层美感，揭示语言设计者的独具匠心。

基于以上观点，全书分为四篇，共计 15 章。每章内容侧重于一组相关的知识和概念。读者在阅读时，能够举一反三，事半功倍。章节按照由简单到复杂、由表层到本质的原则，归入四篇：

第一篇（第 1 章~第 3 章）主要涉及 ActionScript 3.0 的外围知识。第 1 章从脚本语言的基础知识开始，逐步介绍了 ActionScript 3.0 的发展过程、文件结构和运行原理。第 2 章介绍了 ActionScript 3.0 开发常用的三种环境：Adobe 官方的 Flash CS IDE，FlexBuilder 2，以及第三方软件 FlashDevelop 3.0。第 3 章围绕创建“Hello World”程序，向读者介绍了 ActionScript 程序的开发过程及常用开发软件的具体使用方法。

第二篇（第 4 章~第 7 章）讲述了 ActionScript 3.0 中的语言要素。该部分是本书的重点，也是语言的核心。包括语法基础结构，流程控制，面向对象编程和常用对象等内容。本篇是 ActionScript 3.0 开发的基石，为读者进一步学习提供语言元素和基础概念。

第三篇（第 8 章~第 12 章）结合语言的实际应用，介绍简单动画、复杂动画、交互控制、外部数据等语言应用。ActionScript 3.0 是按照类组织的，在解决具体应用时，会涉及多个类。本篇各章节以应用为划分标准，穿插讲解涉及到的相关类。针对不同的应用需求，讲解面向对象编程在解决 Flash 应用问题时的作用。读者阅读本篇后，可以掌握常见应用的实现和核心类的用法。

第四篇（第 13 章~第 15 章）是实战篇。通过编写互动游戏、在线 RSS 新闻阅读器和全屏幕 FLV 视频播放器，涵盖了 ActionScript 3.0 开发的各种知识元素。在构建完整应用程序的过程中，使读者更深刻地掌握开发过程中语言的运用。

本书还介绍了矩阵变换、动画替身等 Flash 编程必备的周边知识和实际技巧，以满足不同读者的阅读需要。

本书的特点

本书以培养实际开发能力为目的，具有以下特点。

- **全面性**：本书不仅介绍了 ActionScript 3.0 的新特性，而且涵盖了语言的所有关键内容。
- **实用性**：紧密联系开发的实际，学以致用。避免了语言的教条罗列，具有很强的实用性。
- **通用性**：解除 ActionScript 3.0 语言与具体软件的绑定，适用于 Flash、Flex 等多种开发环境。
- **渐进性**：重新组织了语言的知识点分布，全书由整体到细节，由浅入深逐步扩展读者的视野。
- **高效性**：将内容整理归类，举一反三。避免相似知识的重复学习，提高学习效率。
- **深入性**：参考了大量 Adobe 内部文档资料，内容深入到语言的深层机制。
- **通俗性**：使用简单易懂的说明语言，配合大量直观的图片示例，将各种复杂概念一一瓦解。
- **整体性**：全书以 OOP 概念和实际开发需求为主线，章节环环相扣，前后呼应，避免了知识单元之间的分散和孤立。

- **独立性：**本书彻底地遵循 ActionScript 3.0 自身逻辑，以 ActionScript 3.0 的角度阐述各种编程概念，帮助读者建立独立完整的开发思维，避免周边语言开发习惯的干扰。
- 配有学习论坛 <http://www.rzchina.net>，读者可以在上面讨论技术，笔者会及时回答读者提问，并提供各种技术文章，帮助你提高开发水平。

格式约定

为了方便读者阅读，本书在编写时使用了以下约定格式。

(1) 代码省略。如果代码较长且出现重复，本书会省略代码中的重复部分。在代码中，将会使用省略号标明省略代码的位置。读者可以根据上下文找到省略的代码，并整理为完整的代码。

例如在下面的代码中，出于分步讲解的需要，if()语句之后的代码被省略了。

```
function animTimerHandler(event:TimerEvent):void {  
    //动画代码编写入口  
    if (animObject1.alpha==1) {  
        fadeStep=-0.05;  
    }  
    .....  
}
```

在随后出现的讲解中，会提到如何完成函数 animTimerHandler()。此时会省略 if()语句，只列出新添加的内容：

```
function animTimerHandler(event:TimerEvent):void {  
    .....  
    animObject1.alpha=fadeInOut(animObject1.alpha, fadeStep);  
}
```

读者可以综合两段代码，得到完整的代码：

```
function animTimerHandler(event:TimerEvent):void {  
    //动画代码编写入口  
    if (animObject1.alpha==1) {  
        fadeStep=-0.05;  
    }  
    animObject1.alpha=fadeInOut(animObject1.alpha, fadeStep);  
}
```

(2) 代码注释。在 ActionScript 3.0 中有两种注释格式：//格式和/**/格式。本书的代码注释也遵从 ActionScript 3.0 的约定。在默认情况下，书中的代码注释并不是代码的组成部分，而是出于讲解需要而添加的。注释应该在代码编译前清除。对于 HTML、XML、MXML、CSS 等代码内容，也套用同样的注释格式：

```
<html>  
    <body>  
        //注释内容  
    </body>  
</html>
```

在实际文件中，不同文件格式的代码可能会具有不同的注释形式，请读者注意区分。

例如，上面的 HTML，在实际使用时应转换为以下形式：

```
<html>
  <body>
    <!-- 注释内容 -->
  </body>
</html>
```

(3) 调试输出。当程序代码会产生调试输出时，一般会使用注释符号将输出列出：

```
var mystr:String;
mystr="A string tracing.";
trace(mystr);
trace(mystr.length);

//output: A string tracing.
//output: 17
```

如果输出信息较多，有可能造成阅读困难。此时本书会将代码输出置于相关的 `trace()` 语句之后，以方便读者。如无特殊说明，`trace()` 语句之后的注释都是该 `trace()` 语句的输出：

```
var mystr:String;
mystr="A string tracing.";
trace(mystr); // A string tracing.
trace(mystr.length); // 17
```

有时为了对比多条输出信息的不同，会视情况将输出信息单独列出。此时上下文会有明确的说明。

适合阅读本书的读者

对于以下人员，本书特别有用。

- ActionScript 3.0 的初学者。
- Flash 交互应用设计师。
- 进行 Flash/Flex 网站开发的代码编写者。
- Web 2.0 网站设计师。
- 对具体语言不熟悉的 RIA 开发者。
- 在 ActionScript 3.0 开发时经常出错的程序员。

阅读本书并不需要读者具备 ActionScript 2.0 经验。设计和美术专业的设计师也可以轻松地阅读本书。对于具备 C++ 或者 Java 等相关语言经验的读者，阅读本书能培养 ActionScript 3.0 的思维模式。

目 录

Contents

第 1 篇 ActionScript 3.0 的外围知识

第 1 章 ActionScript 3.0 综述

2

本章从脚本语言的基础知识开始,逐步介绍了 ActionScript 3.0 的发展过程和运行原理。不仅涉及 ActionScript 3.0, FlashPlayer 和 AVM2 等周边知识,也深入介绍了 Flash SWF 文件的结构和 ActionScript 3.0 在 SWF 文件中的作用。对于 ActionScript 3.0 的编译和执行,本章也有详细的介绍。

1.1 脚本语言基础知识	2
1.1.1 脚本语言的概念	2
1.1.2 脚本和程序	3
1.1.3 脚本语言的特性	3
1.2 Flash 和 ActionScript	4
1.2.1 LEGO 设计模式	5
1.2.2 Flash 和 Action 的起源	5
1.2.3 发展历程	7
1.2.4 ActionScript 3.0 的进步	8
1.2.5 通用性	10
1.2.6 FlashLite	11
1.2.7 ActionScript 3.0 编程工具	12
1.3 Flash 的内部机制	14
1.3.1 Flash 应用的生命流程	14
1.3.2 SWF 文件的 Tag 结构	16
1.3.3 Tag 结构存储实例	17
1.3.4 编译解析	18
1.3.5 AVM2 的解析	19
1.3.6 执行时序	20
1.4 AS3 同 RIA 开发	21
1.4.1 交互程序的进化	22
1.4.2 软件系统体系结构	23
1.4.3 RIA 的诞生及其优势	24
1.4.4 Adobe AIR 简介	26
1.4.5 ActionScript 3.0 与 Adobe AIR 开发	27
1.4.6 AIR 学习步骤	28
1.5 小结	29

本章主要介绍了 ActionScript 3.0 开发常用的三种环境: Adobe 官方的 Flash CS IDE, FlexBuilder 2, 以及第三方软件 FlashDevelop 3.0。经过本章的学习, 读者应该在自己的 PC 上构建至少一种开发环境, 以方便学习后面的编程章节时上机实践。

2.1	Flash CS3 开发环境	30
2.1.1	选择 Flash 版本	30
2.1.2	安装 Flash CS3	31
2.1.3	配置 Adobe AIR 环境	32
2.1.4	Flash CS3 的代码编辑器	33
2.1.5	快捷工具栏	34
2.1.6	专家工具	35
2.1.7	代码调整工具	36
2.1.8	分页式代码标签	37
2.1.9	自定义代码编辑器	37
2.1.10	自动套用格式	38
2.2	Adobe Flex Builder 开发环境	39
2.2.1	系统要求	39
2.2.2	安装准备	40
2.2.3	安装模式	40
2.2.4	安装 Flex Builder 2	41
2.2.5	Flash Player 9 Debug 版	41
2.2.6	完成 Flex Builder 2 的安装	42
2.2.7	Flex Builder 2 集成开发环境	42
2.2.8	Navigator 文件浏览器	43
2.2.9	Flex Builder 中的代码编辑器	43
2.2.10	Outline 提纲浏览器	44
2.2.11	嵌入式 Outline 窗口	45
2.2.12	添加调试面板	46
2.2.13	Flex Builder 2 设计辅助环境	47
2.2.14	自定义 Flex Builder 2	48
2.3	FlashDevelop 开发环境	50
2.3.1	FlashDevelop 3.0 的特色	50
2.3.2	安装 FlashDevelop	51
2.3.3	项目编译测试	52
2.3.4	自定义代码编辑器	53
2.3.5	优化中文字符显示	54
2.3.6	全局自定义类库	55
2.3.7	项目自定义类库	56
2.3.8	设置项目输出	56

2.4 小结	57
--------	----

第3章 创建“Hello World”程序	59
------------------------------	-----------

本章围绕创建“Hello World”程序，向读者介绍了 ActionScript 程序的开发过程，以及常用开发软件的具体使用方法。本章的内容安排本着由简单到复杂，由表面到本质的原则。介绍了如何脱离开发软件，手工实现代码管理和程序编译。帮助读者建立对 ActionScript 3.0 开发流程的整体认识，为进一步研究语言本身打下了基础。

3.1 Flash CS3 创建“Hello World”程序	59
3.1.1 创建 ActionScript 3.0 项目	59
3.1.2 确定代码位置	60
3.1.3 输入 ActionScript 3.0 脚本	61
3.1.4 测试“Hello World”影片	61
3.1.5 使用舞台元件	62
3.1.6 舞台元件与脚本访问	62
3.1.7 控制舞台文本	63
3.1.8 动态创建类型实例	64
3.1.9 操作显示列表	65
3.1.10 ActionScript 3.0 的灵活性	66
3.2 Flex Builder 2 创建“Hello World”程序	67
3.2.1 Flex 程序的构成	67
3.2.2 MXML 的结构	68
3.2.3 MXML 的命名空间	70
3.2.4 MXML 创建和定义组件	71
3.2.5 用户界面同脚本逻辑的分离	72
3.2.6 MXML 的专用组件	72
3.2.7 创建 Flex 项目	73
3.2.8 分析自动代码	74
3.2.9 使用 Flex 快速编写代码	74
3.2.10 编译 MXML 结构的“Hello World”程序	75
3.2.11 在 MXML 中使用 ActionScript 3.0	76
3.2.12 添加标签事件响应	78
3.2.13 MXML 和 Action Script 的执行逻辑	78
3.2.14 ActionScript 的内联、级联和外联	79
3.2.15 使用外联函数	80
3.2.16 可视化 MXML 编辑器	82
3.2.17 复杂代码的编译	84
3.2.18 贯彻 MXML 的分置原则	84
3.3 使用 FlashDevelop 3.0 编写程序	85
3.3.1 新建 FlashDevelop 项目	85
3.3.2 创建 MXML 应用文件	86

3.3.3	代码输入技巧	87
3.3.4	编译和执行 MXML 应用	88
3.3.5	创建外联脚本文件	89
3.3.6	按钮控件及事件响应	90
3.3.7	项目编译和交互测试	91
3.3.8	数字密码输入器	92
3.3.9	界面分析	92
3.3.10	新建项目	94
3.3.11	创建密码框	94
3.3.12	创建按钮	95
3.3.13	建立 ActionScript 3.0 函数	96
3.3.14	整合表示层和逻辑层文件	97
3.3.15	编译和运行程序	98
3.4	手工编译	100
3.4.1	Flex 编译器	100
3.4.2	配置全局路径	102
3.4.3	创建代码文件	104
3.4.4	命令行手工编译	104
3.4.5	手工编译项目	105
3.5	小结	106

第 2 篇 ActionScript 3.0 的语言要素

第 4 章 语法基础结构

108

ActionScript 3.0 是一种基于 ECMA-262 规范的脚本语言，其基本语法构成包括：标识符、关键字、数据类型、运算符和分隔符。这 5 种元素有着不同的语法含义和组成规则，它们互相配合，共同完成 ActionScript 3.0 语言的语意表达。通过本章的学习，可以认识 ActionScript 3.0 中的标识符、关键字、数据类型、运算符和分隔符等语法基础结构。

4.1	标识符	108
4.1.1	标识符的概念	108
4.1.2	标识符命名的技术规则	109
4.1.3	命名约定	111
4.2	关键字	114
4.2.1	表达式	114
4.2.2	语句控制	114
4.2.3	定义	115
4.2.4	属性关键字	116
4.2.5	编译指令	117
4.2.6	区分 IMPORT 和 INCLUDE 关键字	117

051	4.2.7 类的长名称	119
051	4.3 数据类型	120
151	4.3.1 数据类型的概念	120
151	4.3.2 基元值和复杂值	121
151	4.3.3 数据类型的对象封装	122
151	4.3.4 Boolean 数据类型	123
151	4.3.5 int 数据类型	124
151	4.3.6 Null 数据类型	124
151	4.3.7 Number 数据类型	125
151	4.3.8 String 数据类型	127
151	4.3.9 uint 数据类型	127
151	4.3.10 void 数据类型	128
151	4.3.11 无类型数据类型	129
151	4.3.12 复杂数据类型	130
151	4.4 变量和常量	131
151	4.4.1 字面值	131
151	4.4.2 变量及其声明	132
151	4.4.3 变量的赋值	133
151	4.4.4 变量的默认值	134
151	4.4.5 Number 变量和 NaN	134
151	4.4.6 变量的作用域	135
151	4.4.7 变量的后置声明	137
151	4.4.8 常量	138
151	4.5 运算符和功能符	140
151	4.5.1 加法算术运算符	140
151	4.5.2 乘法算术运算符	142
151	4.5.3 一元运算符++和--	144
151	4.5.4 左移位运算符<<	145
151	4.5.5 右移位运算符>>和>>>	147
151	4.5.6 与、或、非的位操作	148
151	4.5.7 位操作与特征位	151
151	4.5.8 特征位的赋值	153
151	4.5.9 特征位的开关操作	154
151	4.5.10 “^”位异或操作符	156
151	4.5.11 高级开关控制	157
151	4.5.12 逻辑运算符	158
151	4.5.13 比较运算符	162
151	4.5.14 赋值运算符	164
151	4.5.15 组合赋值运算符	166
151	4.5.16 功能符	167

011	4.6 类型转换和类型检查	170
051	4.6.1 隐式类型转换	170
051	4.6.2 显式类型转换	171
151	4.6.3 失效转换	172
551	4.6.4 静态类型语法	173
651	4.6.5 语法检查	174
451	4.6.6 编译时类型检查	176
451	4.6.7 运行时类型检查	177
251	4.6.8 数据类型成员的检测	178
751	4.6.9 as 运算符	179
751	4.6.10 强制类型转换	180
851	4.6.11 运算符与隐式类型转换	181
951	4.6.12 数值类型的转换	183
031	4.6.13 布尔类型的转换	186
161	4.6.14 字符串的转换	187
161	4.7 小结	188

第5章 流程控制与函数 189

函数在程序的设计和开发过程中居于极其重要的地位，而使用流程图可以将复杂问题简单化，令抽象过程直观化。本章重点讲解了 ActionScript 3.0 中流程控制和函数使用的相关知识，并介绍了流程图的作用和画法。

751	5.1 条件流程	189
821	5.1.1 条件流程的概念	189
041	5.1.2 if()语句	190
041	5.1.3 if()...else...语句	191
541	5.1.4 三元运算符	193
441	5.1.5 多级条件流程	195
641	5.1.6 if()...else if()...语句	197
741	5.1.7 switch 语句基础用法	199
841	5.1.8 switch 语句常见错误	200
121	5.1.9 条件流程的逻辑控制	203
621	5.2 循环流程	204
421	5.2.1 for 循环	204
621	5.2.2 for()语句执行逻辑	206
721	5.2.3 for...in 循环	207
821	5.2.4 for each...in 循环	209
501	5.2.5 while 循环	211
401	5.2.6 do...while 循环	212
001	5.2.7 break 和 continue	213
741	5.2.8 label 标签与循环嵌套	215

5.2.9	非循环标签	217
5.2.10	标签嵌套	218
5.2.11	循环与作用域	220
5.3	函数	221
5.3.1	函数的基本概念	222
5.3.2	函数定义语句	224
5.3.3	调用函数	226
5.3.4	作用范围和定义位置	227
5.3.5	使用参数	228
5.3.6	可选参数	229
5.3.7	参数的传递	230
5.3.8	函数输出	232
5.3.9	使用返回值	234
5.3.10	函数表达式	235
5.3.11	函数表达式的缺点	238
5.3.12	函数表达式的内存优势	239
5.3.13	函数表达式与编译提升	240
5.3.14	嵌套函数	242
5.3.15	使用 arguments 对象	244
5.3.16	arguments 的方法	246
5.3.17	arguments 与递归函数	248
5.3.18	arguments 与函数重载	251
5.3.19	可变参数	253
5.3.20	函数作用域链	254
5.3.21	函数封包	256
5.3.22	函数对象	257
5.4	流程图	259
5.4.1	程序流程图的概念	259
5.4.2	程序流程图的作用	261
5.4.3	分级流程图体系	263
5.4.4	步骤列表	263
5.4.5	伪代码	264
5.4.6	经典数据流程图的构成	265
5.4.7	指向线与执行框(矩形框)	266
5.4.8	判别框(菱形框)	267
5.4.9	起始框和终止框(圆圈)	267
5.4.10	流程图设计原则	269
5.4.11	常见流程图结构	271
5.4.12	嵌套结构的流程图	272
5.5	小结	274

本章围绕 OOP 这一核心内容,从面向对象编程的基础概念入手,逐步深入。在内容组织上,通用概念和具体实现交织,抽象机制同实际代码并存。先后介绍了 Action Script 3.0 对类的支持和实现,原型对象与抽象类,命名空间和权限控制,包和类文件的管理,类和对象的操作语法等内容。

6.1 对象和类	276
6.1.1 编程模式	276
6.1.2 模式进化	277
6.1.3 OOP 与模块化	279
6.1.4 OOP 开发过程	280
6.1.5 类设计策略	284
6.1.6 类元素的属性定义	286
6.1.7 术语记法	287
6.1.8 类的构造函数	287
6.1.9 继承:父类和子类	288
6.1.10 覆盖和重载	290
6.1.11 多态	293
6.1.12 嵌套类和动态类	294
6.2 面向对象的脚本实现	296
6.2.1 原型对象机制	296
6.2.2 操作原型对象	297
6.2.3 原型链	298
6.2.4 原型继承	300
6.2.5 原型机制下的抽象类	301
6.2.6 抽象类继承	302
6.2.7 操作 AS3 原型对象	304
6.2.8 灵活选择继承机制	306
6.2.9 命名空间的概念	307
6.2.10 创建命名空间	308
6.2.11 应用命名空间	309
6.2.12 引用命名空间	310
6.2.13 关键字 AS3	311
6.3 自定义类	313
6.3.1 自定义类的作用	313
6.3.2 定义 ActionScript 3.0 类	314
6.3.3 类属性关键字	316
6.3.4 类体	317
6.3.5 属性和方法	318
6.3.6 创建类成员	319